Go Complexities

Abdallah Saffidine, Olivier Teytaud, Shi-Jim Yen

Go Complexities

Abdallah Saffidine[1], Olivier Teytaud[2], Shi-Jim Yen[3]

[1] CSE, The University of New South Wales, Sydney, Australia.
[2] Tao team, Inria, LRI, Univ. Paris-Sud, France.
[3] Ailab, CSIE, National Dong Hwa University, Taiwan.

Abstract. The game of Go is often said to EXPTIME-complete. The result refers to classical Go under Japanese rules, but many variants of the problem exist and affect the complexity. We survey what is known on the computational complexity of Go and highlight challenging open problems. We also propose a few new results. In particular, we show that Atari-Go is PSPACE-complete and that hardness results for classical Go carry over to their Partially Observable variant.

1 Introduction

The 2000's and 2010's have seen a huge progress in computer Go, with the advent of Monte Carlo Tree Search (MCTS) [3], sequence-like simulations [7], rapid action value estimates [6], and human expertise included in the tree search. The game of Go is rich of teminologies; there are *semeais* which are hard for MCTS algorithms, *ladders* which connect remote parts of the board, *Ko fights* which are at the heart of Robson's proof of EXPTIME-hardness for Go with Japanese rules, and *Tsumegos* (problems of varying difficulty, studied both by beginners and experts). It is also rich of variants, such as Killall Go and the partially observable variant *Phantom Go*.

The complexity of problems for a game is usually formalized as follows. A position is any particular state of the game, *i.e.*, in the case of Go with Japanese rules a board equipped with stones. We consider a family P of initial positions, and we consider the following decision problem:

Sure win problem: Is a given position $p \in P$ a sure win for Black assuming perfect play?

The complexity of a game is the complexity of this decision problem. This is well defined for fully observable (FO) games, as long as rules are formally described, and all information pertaining to a state is included in the input.

- For Go with Japanese rules, not all situations are completely formalized. Fortunately, the rules are not ambiguous for a large subset of positions, and we can still work on the complexity of Go with Japanese rules.
- Chinese rules present a distinct problem: some necessary information is not included in the position. Indeed, because cycles are forbidden by the superko rule, one must remember past positions and avoid them. We will consider the decision problem associated to positions, considering there is no history of forbidden past state. This is the classical considered setting, *e.g.* in [8].

For partially observable variants, we still assume a fully observed input state, which is then played with its partially observable setting. The decision problem is not a sure win anymore but rather winning with a sufficiently high likelihood:

> **Threshold problem:** Given a fully observable position $p \in P$ and a rational number $c \in [0, 1]$, is there a strategy for Black (possibly but not necessarily randomized) so that for all White strategies, Black wins with probability at least c when the game starts in position p?

2 Rules, variants, and terminology

At the beginning of the game, the board is empty. Players, starting with Black, take turns putting a stone of their color on an free intersection. When a maximum group of 4-connected stones of a same color has no *liberty* (*i.e.*, does not touch any free intersection), then it is removed from the board — this is termed *capture*. Suicide, *i.e.*, playing a stone that would be immediately captured, is forbidden. Playing a move which goes back to the previous situation is forbidden; this is the *Ko* rule. In some versions, any move which goes back to an earlier position is forbidden; this is the *Superko* rule. The game terminates when no players want/can play. The *score* of a player, according to Chinese rules, is then the number of stones on the board, plus the surrounded empty locations. As Black plays first, White is given a bonus, termed *Komi*. The player with the greatest score is the winner.

Black's inherent first move advantage may skew games between equally skilled opponents. It is therefore traditional to give White a number of bonus points called *Komi* as a compensation. The larger the Komi, the easier the game for White. A Komi of 7 is considered fair, but to prevent draws, non-integer Komi such as 6.5 and 7.5 are regularly used. Games between unequally skilled opponents need to use a balancing scheme to provide a challenge for both players. The *handicap* mechanism is popular whereby Black is given a specified number of free moves at the start of the game, before White plays any move. The larger the handicap, the easier the game for Black.

Chinese and Japanese rules. In most practical settings, there is no difference between using Chinese or Japanese rules. In mathematical proofs, however, it makes a big difference, as the latter allow cycles. Cycles of length 2 (termed *ko*) are forbidden in all variants of the rules, but longer cycles (termed *superko*) are forbidden in Chinese rules, whereas they are allowed in Japanese rules.

In Japanese rules, when a cycle occurs, the game is considered as a draw and replayed.[4] More than 20 pro games draw with superko are known in the world since 1998. On 2012, September 5th, in the Samsung world cup, the game Lee Shihshi - Ku Li featured a quadruple Ko, leading to a loop in the game, which was replayed.

Ladders are an important family of positions in Go. They involve a growing group of stones, *e.g.*, white stones; Black is trying to kill that group, and White is

[4] See the detailed rules at `http://www.cs.cmu.edu/~wjh/go/rules/Japanese.html`

trying to keep it alive by extending it. At each move, each player has a limited set of moves (one or two moves usually) which are not immediate losses of the ladder fight. An interesting result is that ladders are PSPACE-hard [9]: one can encode geography (PSPACE-complete) in planar-geography, and planar-geography in Go. So, Go, even restricted to ladders, is PSPACE-hard. This result is less impressive than the EXPTIME-hardness, but the proof is based on moderate size gadgets which can almost really be reproduced in a Go journal.

[4] proved that *Tsumegos*, in the restricted sense of positions in which, for each move until the problem is over, one player has only one meaningful move and the other has 2, are NP-complete. This means that there is a family of Tsumegos (with the previous definition, from [4]) for which the set of positions which are a win for Black, in case of perfect play, is NP-complete.

2.1 Go Variants.

Killall is a variant of Go in which Black tries to kill all groups of the opponent and White tries to survive. The Killall Go variant can be seen as a special case of Classic Go with appropriately set Komi and handicap. The Komi is almost the size of the board, so that White wins if and only if she has at least one stone alive at the end of the game. The handicap is large enough that the game is approximately fair. For example, on a 9×9 board, the Komi would be set to 80.5 and the handicap would be 4 stones; on a 19×19 board, the Komi would be 360.5 and the handicap between 17 and 20 stones.

Atari Go is another variant, in which the first capture leads to a victory. Therefore, Komi has no impact. Atari Go is usually played as a first step for learning Go.

Phantom Go is a partially observable variant of Go. Given a fully observable game G, *Phantom G* denotes the following variant. A referee (possibly a computer is the referee) takes care of the board, and players only see their stones on their private version of the board. That is, White is only informed of the position of White stones, whereas Black is only informed of the position of Black stones. Moves are not announced, so even with perfect memorization, it is usually not possible to know where are the opponent stones. Illegal moves should be replayed. Depending on variants, the player can get additional information, such that captures, and possibly location of captured stones.

Blind Go exactly follows the rules of Go, except that the board is not visible; players must keep the board in mind. In case of illegal move due to memorization error, they should propose another move. *One-color Go* is an easier version: there are stones on the board, but they all have the same color; players should memorize the "true" color of stones. In these variants, the computational complexity is the same as in the original game; contrarily to Phantom Go, the difference with Go boils down to a memory exercise.

3 Results in fully observable variants

3.1 Japanese rules

[10] has proved the EXPTIME-hardness of a family P of Go positions for which the rules are fully formalized. This is usually stated as EXPTIME-completeness of Go. The point is that Go with Japanese rules is not formally defined; there are cases in which the result of the game is based on precedents or on a referee decision rather than on rules (confusing examples are given in `http://senseis.xmp.net/?RuleDisputesInvolvingGoSeigen`, and precedents are discussed in `http://denisfeldmann.fr/rules.htm#p4`).

However, FO games are EXPTIME-complete in general, unless there are tricky elements in the evaluation of the result of the game (more formally: if evaluating a final state can not be done in exponential time) or in the result of a move (more formally: the board after a move is played can be computed in exponential time); therefore, we can consider that Go is EXPTIME-complete for any "reasonable" instantiation of the Japanese rules, *i.e.*, an instantiation in which (i) the situations for which the rules are clear are correctly handled (ii) deciding the consequences of a move and who has won when the game halts can be done in exponential time.

The original proof by Robson is based on Ko fights, combined with a complex set of gadgets that correspond to a ladder [10]. More recent work has shown that these ladder gadgets could be simplified [5].

3.2 Chinese rules

Go with Chinese rules is different. The same Ko fight as in Robson's proof can be encoded in Chinese rules, but the result is different (at least if humans follow the superko rules forbidding cycles, which is not that easy in some cases). The PSPACE-hardness is, on the other hand, applicable with Chinese rules as well as with Japanese rules; therefore, one might, at first view, believe that Go with Chinese rules is either PSPACE-complete or EXPTIME-complete. However, the state space with Chinese rules is much bigger than the size of the apparent board: one must keep in memory all past positions, to allow avoiding cycles. As a consequence, Go with Chinese rules is not subject to the general EXPTIME result; and EXPSPACE is the current best upper bound.

Theorem 1 (folklore result). *Go with Chinese rules is in* EXPSPACE.

Proof. Extend the state with an (exponential size) archive of visited states. This augmented game is acyclic. Therefore it is solved in polynomial space (by depth first search of the minimax tree) on this exponential size representation; this is therefore an EXPSPACE problem.

This is widely known and we do not claim this as a new result. Lower bounds for games with a no-repeat condition are typically much more difficult to obtain, but the general case is EXPSPACE-hard [8, Section 6.3]. There is no EXPTIME-hardness of EXPSPACE-hardness result for the specific case of Go with Chinese rules. So the actual complexity of Go with Chinese rules is open and might lie anywhere between PSPACE and EXPSPACE. A nice consequence in complexity theory is that if Go with Japanese rules is harder than Go with Chinese rules (in a computational complexity perspective), then EXPTIME (where we find Go with Japanese rules) is different from PSPACE.

3.3 Killall Go variant

A detailed proof of complexity for Killall Go is beyond the scope of this paper; but we give a few hints in this direction. A key component for applying classical results (such as the EXPTIME-hardness with Japanese rules or the PSPACE-hardness with Chinese rules) is to rewrite

1. a big group, the life or death of which decides the result of the game (this is a key component for all proofs); this big group can make life only through a ladder.
2. the Ko-fight (necessary for the EXPTIME-hardness of Japanese Go).
3. the ladder components from [5], or the ones from [10], (both are equivalent, the ones from [5] are simpler), necessary for both the PSPACE-hardness of Chinese Go and the EXPTIME-hardness of Japanese Go.

These components should be adapted to the Killall Go setting. The first part of this, number 1 above, consists in designing a position in which winning the game boils down to winning the ladder, thanks to a big group which must live thanks to the ladder. First, we need a group which will live or die, only depending on the ladder. This is easily obtained as shown in Fig. 1.

The Ko fight is also not a problem; there is no room for making life around the Ko fight, so Killall Go and Go are equivalent for this part of the game. Then, we must adapt the gadgets for the ladder itself. The difficult point is to ensure that there is no room elsewhere in the board in which White might make life, out of the line of play used in the widgets. The principles are as follows:

– Fill all empty spaces with strong Black stones, which can not be killed. We only have to keep two empty points beside each point of the ladder path; we can fill all other parts of the board with black stones and their eyes.
– Since the ladder path is thin and the surrounding Black stones are very strong, White can not make two eyes even if she plays three continuous white stones in a row. We get ladders as in e.g. Fig. 2.

We believe this statement can be made rigorous by a long and tedious formalization; this is however a very long exercise, beyond the scope of this work, so we only have the following conjecture:

Conjecture 1. Killall Go with Japanese rules is EXPTIME-hard. Killall Go with Chinese rules is PSPACE-hard and EXPSPACE.

Note that Killall Go with Chinese (resp. Japanese) rules is in EXPSPACE (resp. EXPTIME) because it forms a subset of decision problems for Classical Go.

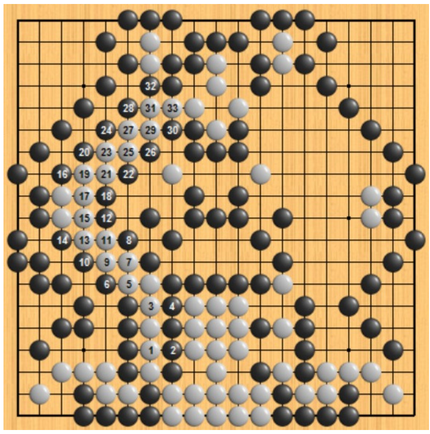


Fig. 1. A position in which White wins (in a Killall Go sense) if and only if she can win the ladder.

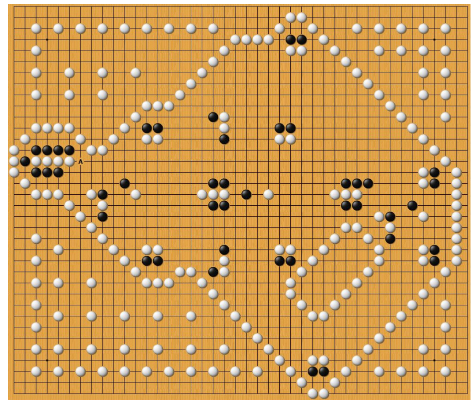


Fig. 2. White to play. A ladder, adapted to Killall Go: the left-most Black group can make life by killing in this ladder. Black can not make life without winning the ladder.

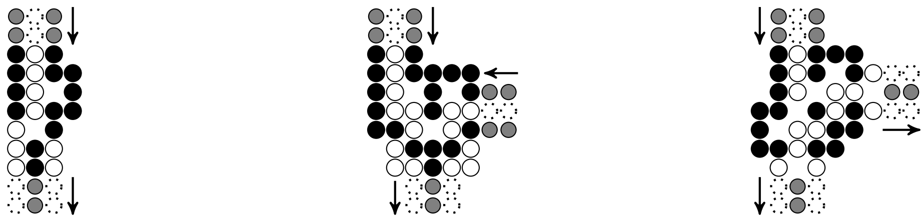
3.4 Atari Go

There is no capture until the end of a game of Atari-Go and each turn a new stone is added to the board, therefore the game is polynomially bounded and is in PSPACE. To prove hardness we will reduce from the two-player game Generalized Geography. The variant we use assumes a planar bipartite graph of degree at most 3 and is PSPACE-hard [9].

We can partition the set of vertices of a planar bipartite graph of degree 3 based on the indegree, the outdegree, and the color of the vertex. We can restrict ourselves to considering the following 6 types of vertices: White vertex with indegree 1 and outdegree 1 (W1-1 vertex for short), W2-1, W1-2, B1-1, B2-1, and B1-2. The Atari Go gadgets for each White vertex type are presented in Fig. 3, and the gadgets for Black vertex types are the same with the color reversed.

The gadgets comprise *interior groups*, *exterior groups*, and *link groups*. The interior groups only appear in 1-2-vertex gadgets and initially contain 1 or 3 stones. The exterior groups all have a large number of liberties and will never be under any capturing threat. The link groups serve as links between the different gadgets. Each link group is flanked on each side by an exterior group of the opposite color, and the three groups take the shape of a corridor so as to imitate the path of the Generalized Geography edge. To each edge in the Generalized Geography instance correspond a single link group in the Atari Go instance.

Our construction ensures that initially, each link group has two *end liberties* in the vertex gadget it arrives in and one *starting liberty* in the vertex gadget it departs from. As such, before the start of the game, every group on the board has at least 3 liberties, except for single-stone interior groups. Playing in the



(a) 1-1 vertex (b) 2-1 vertex (c) 1-2 vertex

Fig. 3. Gadgets reducing Generalized Geography on bipartite planar graphs of degree at most 3 to Atari-Go. Each gadget represent a type of white vertex, the black vertices can be represented by flippling colors.

starting liberty of a link group simulates selecting the corresponding edge in Generalized Geography, we call it *attacking* this link group.

Lemma 1. *If the opponent just attacked a given link group, it is a dominant strategy for the player to extend this group by playing in one of the two end liberties.*

In the case of the 1-1 and 2-1 gadgets, one of the end liberties is only shared with an opponent's exterior group, so it is dominant to play in the other end liberty. This remark and Lemma 1 ensure that it is dominant for both player to continue simulating Generalized Geography on the Atari Go board.

Theorem 2. *Atari Go is* PSPACE*-complete.*

Proof. The branching factor and the game length are bounded by the size of the board so the problem belongs to PSPACE. Generalized Geography on planar bipartite graphs of degree 3 is PSPACE-hard and can be reduced to Atari Go as we have shown above. Therefore, Atari Go is PSPACE-hard as well.

While the complexity of Atari Go is the same as that of Ladders, a different proof was needed because one of the ladder gadgets involves and intermediate capture.

4 Phantom Go

Go is usually FO, but variants in which the opponent's moves are not observed have been defined. In order to preserve the consistency of games, when a move is illegal (typically, but not only, because it is played on an opponent's stone) then the player is informed, and the move is replayed. Detailed rules differ, but players are always informed when their stones are captured. A consequence is that Ko fights can be played as in the FO game.

Phantom Go is a nice challenge, in particular because the best algorithms are somehow simple; basically, the pseudo-code of the best algorithms for playing in a situation S is as follows, for some parameter N:

– For each possible move m, repeat N times:
  - let K be the number of unknown opponent stones;
  - randomly place the K unknown opponent stones on the board;
  - perform a Monte Carlo simulation
– Play the move m with best average performance.

In particular, this is not a consistent approach; even with N infinite, such an algorithm is not optimal. Nonetheless it usually performs better than more sophisticated approaches [2].

4.1 Lower bounds on Phantom Go complexity

We here present lower bounds on Phantom Go complexities derived by adapting the proofs in the FO case. There are two sets of gadgets we would like to use in the phantom framework:

– The Ko fight defined in Robson's work[10]. The situation here is easy, because, in Phantom Go rules, captures are visible. Ko fights involve one capture per move; therefore, they are played exactly as if it was standard Go.
– The ladder gadgets, either from [10] or [5]. These gadgets are necessary both for adapting Robson's proof to the Phantom Go case (there is more than a Ko fight in [10], there is also a ladder), and directly for the PSPACE hardness of ladders in the Phantom Go case. We reproduce the gadgets ladders in Fig. 4 for convenience [5, Figure 2].

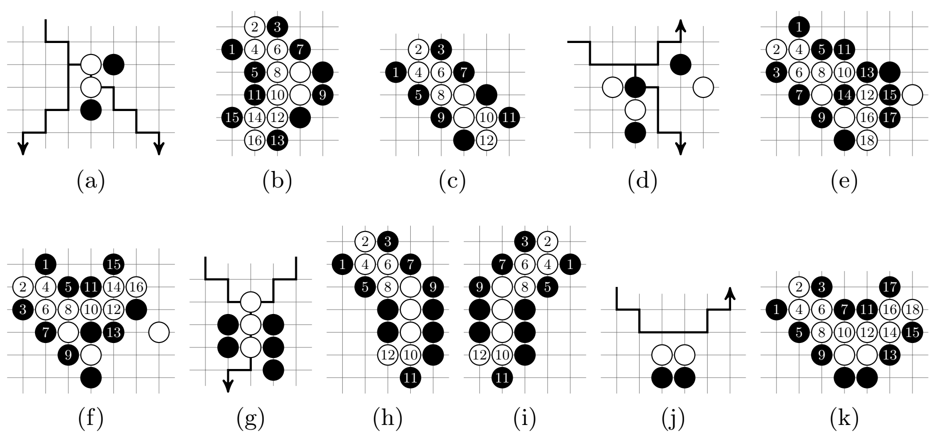


Fig. 4. Gadgets to build a PSPACE-complete family of ladder problems [5, Fig. 2]. We show that the same family of ladder problems is PSPACE-complete in Phantom Go.

Theorem 3. *Ladders are* PSPACE*-hard also in Phantom Go with both Chinese and Japanese rules. Go is* EXPTIME*-hard in Phantom Go with Japanese rules.*

This holds for the existence of a sure win for Black, and therefore also for the threshold problem.

Proof. We consider the problem of the existence of a sure win for Black.

Preliminary remark: Black has a sure win, if and only if Black has a deterministic strategy for winning surely against the White player playing uniformly at random. Therefore, Black has a sure win even if the White player, by chance, plays exactly as if White could see the Board.

The proof consists in using the same positions as in [5], and showing that strategies used in [5] can be adapted to the Phantom Go case, with the key property that optimal strategies in [5] are adapted to optimal policies in the Phantom Go case. This implies that we can adapt ladders in Phantom Go. Thanks to the remark above (Ko fights are played in Phantom Go with the same line of play as in Go) we can also play Ko fights. So, simulating ladders leads both to PSPACE-hardness (for both Chinese and Japanese rules) and to EXPTIME-hardness (for Japanese rules, because we simulate a Ko fight for which the hardness proof exists only in the case of Chinese rules).

So, let us consider ladders, and let us simulate them in Phantom Go. In the (fully observable) Go version of the game, the line of play is as follows.

Each player is forced to play the line of play described in Fig. 4, otherwise she loses the game [5]. The *Black Choice* 4a gadget leads to 4b or 4c, up to Black. *White Choice* 4d leads to 4e or 4f, up to White. *Join* leads to 4g, in case the ladder comes as in 4h or as in 4i. *Mirror* leads to 4k. In all cases, there are at most two choices for each player at each move.

Case 1: when White has a forced move. Let us consider the case in which, in Go, White has only one move M which is not an immediate loss. In the posititons proposed in [5] this happens when the ladder just propagates, without any choice, including the mirror and the join.

Let us show that it is also the case in Phantom Go. This case is the easiest: the killing move for Black is precisely the move M that White should play. So Black can just try to play M. If M kills, Black wins. Otherwise, Black is informed that White has played M, since the rules of Phantom Go state that in case a move is illegal for Go it should be replayed elsewhere.

Case 2: when Black has a forced move, *i.e.*, a Move such that all other moves immediately make the White ladder alive. By the preliminary remark, if Black does not play this move, it does not have a sure Win. This case is concluded.

Case 3: when White has a Choice. Let us consider the case in which, in Go, White has two choices. This is the case in "White Choice" gadgets. Let us show that it is also the case in Phantom Go, and that Black can guess which move White has played. Let us consider that White plays, one of the two choices displayed in Fig. 4e and 4f (Fig. 4, move 12 in 4e and 4f respectively). Let us show that Black can guess which move and reply as in 4e and 4f, even in the Phantom Go case. This is easy because the move number 12 in 4e is a capture.

So if White plays this move, Black is informed, and we are back at the line of play of 4e. Otherwise, Black can just check that White has played the move 12 in 4f by trying to play it: if White had not actually played that move, then this captures the ladder and Black has won.

Case 4: when Black has a Choice. Let us consider the case in which, in the FO line of play, Black has two choices. By the preliminary remark, Black wins surely only if it wins surely if White plays with knowledge of Black's move, so we can consider the case of omniscient White. If Black does not play one of the two moves, White makes life in the FO line of play. If Black plays one of the two moves, we can consider the case in which White has made a correct assumption on which move Black has played.

4.2 Upper bounds on Phantom Go complexity

Japanese rules Given that some partially observable deterministic two-player games can be undecidable even when restricted to a finite state space [1], it is not even clear that Phantom Go with Japanese rules is decidable at all. The best bound we have is the following:

Theorem 4. *Phantom Go with Japanese rules is in 0', the set of decision problems which can be decided by a Turing machine using the halting problem as an oracle.*

Proof. Thanks to Theorem 5 in [1], the following machine halts if and only if Black has a strategy for winning with probability greater than c, in Phantom Go with Japanese rules:

- $K = 1$
- while (true)
  - Solve exactly the game up to K time steps, starting in a given position p.
  - if Black can win with probability $> c$, then halt.
  - Otherwise, $K \leftarrow 10K$

Then, the following machine with the halting problem as oracle can solve the decision problem of the present theorem:

- Input: a position p.
- Output:
  - output *yes* if the machine above halts.
  - output *no* otherwise.

This yields the expected result.

This implies that finding the optimal move is also computable in 0'. So, as a summary: Phantom Go with Japanese rules is EXPTIME-hard and in 0'.

Chinese rules We propose an upper bound for Phantom Go, in the case of Chinese rules. Due to Chinese rules, there is a superko rule: it is forbidden to have twice the same move. We claim the following:

Theorem 5. *Phantom Go with Chinese rules is in* 3EXP*, the set of decision problems which can be solved in* $2^{2^{2^{\mathsf{poly}(n)}}}$*, where* $\mathsf{poly}(n)$ *is a polynomial function.*

Proof. We consider games of Phantom Go with Chinese rules, on an $n \times n$ board, and we compute upper bounds on the number of board positions, B, the maximum length of a game, L, the number of distinct histories, N, and the number of pure (i.e., deterministic) strategies, P. We first have $B = 3^{n \times n}$, since any intersection can be empty, black, or white.

The length of a game is at most $L = 3(1 + n \times n)B$, where the 3 factor indicates whether players have passed zero, one, or two times in the current position. The $n \times n + 1$ factor arises because a player can play again when their move is rejected in Phantom Go, for each "real" move, there are at most $n \times n + 1$ trials (including pass), which can be (i) accepted or (ii) rejected as illegal. Finally the B factor is due to the superko rule: each position is allowed at most once.

One move leads to $(n \times n + 1)$ possibilities (including pass). Then, the player observes some information, which is of size at most $1 + 2^{n \times n}$; this is the number of possible captures, plus one for the "illegal" information. Therefore, an upper bound on the number of histories is $N = ((n \times n + 1)(1 + 2^{n \times n}))^L$.

Pure strategies are mappings from histories to actions; there are therefore at most $P = (1 + n \times n)^N$ pure strategies. The $1 + n \times n$ stands for the $n \times n$ standard moves, plus a possibility of pass.

Substituting the B, L, and N for their values, we obtain that the number of pure strategies is at most a tower of 3 exponentials in n. Solving the input Phantom Go position can therefore be reduced to solving a normal-form game of size triply exponential in n. Two-player zero-sum games in normal-form can be solved in time polynomial in the size of the matrix, for instance by solving the corresponding linear program, so Phantom Go can be solved in time triply exponential in n.

5 Conclusion

We (i) Surveyed the state of the art in the complexity of Go, (ii) Proved a new result for Atari-Go, (iii) Proposed (without formal proof) the extensions of these results to Killall Go variants, (iv) And proved lower and upper bounds for Phantom Go. The main open problems are (i) The decidability of Phantom Go with Japanese rules (because undecidability would be the first such result for a game played by humans), (ii) The complexity of Go with Chinese rules (because the gap between the lower and upper bounds is huge, more than for any other game), (iii) The complexity of Phantom Go with Chinese rules. Table 1 summarizes the known results on the complexity of Go and its variants. Upper bounds for Japanese rules assume that we only consider reasonable formalizations of the rules and rule out ambiguous positions.

Table 1. Summary of the complexity of Go-related problems. Results with a * have not been fully formalized yet and should be considered as conjecture only. New results are highlighted in boldface.

Rules	Variant	Lower bound	Upper bound
	Atari-Go	**PSPACE-hard**	**PSPACE**
Chinese	Classic	PSPACE-hard	EXPSPACE
	Phantom	**PSPACE-hard**	**3EXP**
	Killall	PSPACE-hard*	**EXPSPACE**
Japanese	Classic	EXPTIME-hard	EXPTIME
	Phantom	**EXPTIME-hard**	**0'** (decidability unsure)
	Killall	EXPTIME-hard*	**EXPTIME**

Acknowledgments

We are grateful to Tristan Cazenave for fruitful discussions around Phantom Go. The first author was supported by the Australian Research Council (project DE 150101351).

References

1. Auger, D., Teytaud, O.: The frontier of decidability in partially observable recursive games. International Journal on Foundations of Computer Science 23(7), 1439–1450 (2012)
2. Cazenave, T., Borsboom, J.: Golois wins Phantom Go tournament. ICGA Journal 30(3), 165–166 (2007)
3. Coulom, R.: Efficient selectivity and backup operators in Monte-Carlo Tree Search. In: van den Herik, H., Ciancarini, P., Donkers, H. (eds.) Computers and Games, Lecture Notes in Computer Science, vol. 4630, pp. 72–83. Springer Berlin / Heidelberg (2007)
4. Crâşmaru, M.: On the complexity of Tsume-Go. In: van den Herik, H.J., Iida, H. (eds.) Computers and Games, Lecture Notes in Computer Science, vol. 1558, pp. 222–231. Springer Berlin Heidelberg (1999)
5. Crâşmaru, M., Tromp, J.: Ladders are PSPACE-complete. In: Marsland, T., Frank, I. (eds.) Computers and Games, Lecture Notes in Computer Science, vol. 2063, pp. 241–249. Springer Berlin / Heidelberg (2001)
6. Gelly, S., Silver, D.: Combining online and offline knowledge in UCT. In: ICML '07: Proceedings of the 24th international conference on Machine learning. pp. 273–280. ACM Press, New York, NY, USA (2007)
7. Gelly, S., Wang, Y., Munos, R., Teytaud, O.: Modification of UCT with patterns in Monte-Carlo Go. Rapport de recherche INRIA RR-6062 (2006), `http://hal.inria.fr/inria-00117266/en/`
8. Hearn, R.A., Demaine, E.D.: Games, Puzzles, and Computation. A K Peters (Jul 2009)
9. Lichtenstein, D., Sipser, M.: Go is polynomial-space hard. Journal of the ACM 27(2), 393–401 (1980)
10. Robson, J.M.: The complexity of Go. In: IFIP Congress. pp. 413–417 (1983)